Non-Bayesian Additive Regularization for Multimodal Topic Modeling of Large Collections

Konstantin Vorontsov
Yandex, Moscow Institute
of Physics and Technology
voron@yandex-team.ru

Oleksandr Frei
Schlumberger
Information Solutions
oleksandr.frei@gmail.com

Anastasia Yanina
Moscow Institute of Physics
and Technology
yanina.anastasia.mipt@gmail.com

Murat Apishev
Moscow State University
great-mel@yandex.ru

Peter Romov
Yandex
peter@romov.ru

Marina Suvorova
Moscow State University
m.dudarenko@gmail.com

ABSTRACT

Probabilistic topic modeling of text collections is a powerful tool for statistical text analysis based on the preferential use of graphical models and Bayesian learning. Additive regularization for topic modeling (ARTM) is a recent semi-probabilistic approach, which provides a much simpler inference for many models previously studied only in the Bayesian settings. ARTM reduces barriers to entry into topic modeling research field and facilitates combination of topic models. In this paper we develop the multimodal extension of ARTM approach and implement it in BigARTM open source project for online parallelized topic modeling. We demonstrate the ability of non-Bayesian regularization to combine modalities, languages and multiple criteria to find sparse, diverse, and interpretable topics.

Keywords

Probabilistic Topic Modeling, Probabilistic Latent Sematic Analysis, Latent Dirichlet Allocation, Additive Regularization of Topic Models, Stochastic Matrix Factorization, EM-algorithm, BigARTM.

1. INTRODUCTION

Topic modeling is a rapidly developing branch of statistical text analysis [2]. Topic model reveals a hidden thematic structure of a text collection and finds a compressed representation of each document in terms of its topics. Practical applications of topic models include information retrieval, classification, categorization, summarization and segmentation of texts. Topic models are increasingly used for non-textual and heterogeneous data including signals, images, video and networks. More ideas, models and applications are outlined in the survey [7].

From a statistical perspective, a probabilistic topic model (PTM) defines each topic by a multinomial distribution over words, and then describes each document with a multinomial distribution over topics.

Modern literature on topic modeling offers hundreds of models adapted to different situations [7]. Nevertheless, most of these models are too difficult for practitioners to quickly understand, adapt and embed into applications. This leads to a common practice of tasting only the basic out-of-date models such as *Probabilistic Latent Semantic Analysis*, PLSA [12] and *Latent Dirichlet Allocation*, LDA [4]. Most practical inconveniences are rooted in Bayesian learning, which is the dominating approach in topic modeling.

Bayesian learning is very powerful and general theoretical framework, for which topic modeling is one of example applications. Bayesian inference is elegant when conjugate priors are used. However, the Dirichlet conjugate prior is not always a better choice from the linguistic point of view. In particular, it conflicts with natural assumptions of sparsity. Better motivated non-conjugate priors require a laborious mathematical work and lead to intricate learning algorithms. The development of combined and multi-objective topic models also remains a challenging task in the Bayesian approach. An evolutionary approach to multi-objective Bayesian topic modeling has been proposed in [14], but it seems to be computationally infeasible for large text collections. Until now, there was no freely available software to combine topic models.

From an optimization perspective, topic modeling can be considered as a special case of approximate stochastic matrix factorization. To learn a factorized representation of a text collection is an ill-posed problem, which has an infinite set of solutions. A typical regularization approach in this case is to impose problem-specific constraints in a form of additive terms in the optimization criterion.

Additive Regularization of Topic Models (ARTM) is a semi-probabilistic approach based on classical (non-Bayesian) regularization [31]. In ARTM a topic model is learned by maximizing a weighted sum of the log-likelihood and additional regularization criteria. These criteria are not required to be log-priors or even to have a probabilistic sense. The optimization problem is solved by a general regularized expectation-maximization (EM) algorithm, which can be easily applied to any combination of regularization crite-

CIKM CIKM 2015 Workshop on Topic Models: Post-Processing and Applications

ria. The non-Bayesian regularization provides a much simpler inference for many topic models previously studied only in the Bayesian setting [33, 32]. In particular, the LDA model can be alternatively understood as a smoothing regularizer that minimizes Kullback–Leibler (KL) divergence of each topic distribution with a fixed multinomial distribution. The maximization of the KL-divergence naturally leads to sparsing [33]. This possibility is difficult to see from the Bayesian perspective, thereby all Bayesian approaches to sparsing are much more complicated [26, 35, 16, 10, 5].

ARTM makes topic models easier to design, to explain, to infer, and to combine, thereby reducing barriers to entry into topic modeling research field.

In this paper we develop the multimodal extension if ARTM approach and contribute its parallel online implementation into BigARTM open source projet.

Multimodal data has become increasingly important in many application areas. Large collections of data coming from the web consist of heterogeneous linked data. Typically, texts are accompanied by images, audio or video clips, usage data, metadata containing authors, links, date-time stamps, etc. In these cases documents are considered as multimodal containers, for which words are the elements of only one of multiple modalities. All modalities are useful for determining more relevant topics, and, vice-versa, topics are useful for crossmodal retrieval, making recommendations for users or making predictions when data of some modalities are missing. We introduce the multimodal additively regularized topic model with an arbitrary number of modalities and generalize the regularized EM-algorithm for this case.

Online algorithms have proven to be the fastest for the large document collections, including those arriving in a stream. Online algorithms are now available for PLSA [1], LDA variational inference [11], LDA stochastic inference [18], and some other topic models. We show that the online algorithm is not necessarily associated with a particular type of model, nor a particular type of inference, but only with a certain reorganization of steps in the EM-like iterative process. Our online algorithm remains the same for any combination of regularizers and any number of modalities.

The rest of the paper is organized as follows. In section 2 we introduce notation and definitions of topic modeling and ARTM. In section 3 we introduce a multimodal topic modeling for documents with additional discrete metadata. In section 4 we generalize the fast online EM-algorithm [11] for multimodal ARTM and discuss some details of its parallel implementation in BigARTM library. In section 5 we report results of our experiments on large datasets.

2. ADDITIVE REGULARIZATION FOR PROBABILISTIC TOPIC MODELS

Let D denote a finite set (collection) of texts and W denote a finite set (vocabulary) of all terms from these texts. Each term can represent a single word or a key phrase. Each document $d \in D$ is a sequence of terms from the vocabulary W. Assume that each term occurrence in each document refers to some latent topic from a finite set of topics T. Text collection is considered to be a sample of triples (w_i, d_i, t_i), $i = 1, \dots, n$, drawn independently from a discrete distribution $p(w, d, t)$ over the finite probability space $W \times D \times T$. Terms w_i and documents d_i are observable variables, while topics t_i are latent variables.

The topic model of Probabilistic Latent Semantic Analysis, PLSA [12] explains the terms probabilities $p(w \,|\, d)$ in each document $d \in D$ by a mixture of term probabilities for topics and topic probabilities for documents:

$$p(w \,|\, d) = \sum_{t \in T} p(w \,|\, t)\, p(t \,|\, d) = \sum_{t \in T} \phi_{wt}\theta_{td},\ w \in W.$$

This representation follows immediately from the law of total probability and the assumption of conditional independence $p(w \,|\, t) = p(w \,|\, d, t)$, which means that each topic generates terms regardless of the document.

The parameters $\theta_{td} = p(t \,|\, d)$ and $\phi_{wt} = p(w \,|\, t)$ form matrices $\Theta = \bigl(\theta_{td}\bigr)_{T \times D}$ and $\Phi = \bigl(\phi_{wt}\bigr)_{W \times T}$. These matrices are *stochastic*, that is, their vector-columns represent discrete distributions. The number of topics $|T|$ is usually much smaller than $|D|$ and $|W|$.

To learn parameters Φ, Θ from the collection we maximize the log-likelihood:

$$\mathscr{L}(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in W} n_{dw} \ln p(w \,|\, d) \to \max_{\Phi, \Theta},$$

where n_{dw} is the number of occurrences of the term $w \in W$ in the document d.

Following the ARTM approach, we introduce r additional criteria $R_i(\Phi, \Theta)$, $i = 1, \dots, r$, called *regularizers.* We would like to maximize them separately, but the maximization of their linear combination with nonnegative *regularization coefficients* ρ_i is technically more convenient:

$$R(\Phi, \Theta) = \sum_{i=1}^{r} \rho_i R_i(\Phi, \Theta) \ \to\ \max_{\Phi, \Theta}.$$

Then we add a regularization term $R(\Phi, \Theta)$ to the log-likelihood and solve a constrained multicriteria optimization problem via scalarization:

$$\mathscr{L}(\Phi, \Theta) + R(\Phi, \Theta) \to \max_{\Phi, \Theta}; \qquad (1)$$

$$\sum_{w \in W} \phi_{wt} = 1,\ \phi_{wt} \geq 0; \qquad \sum_{t \in T} \theta_{td} = 1,\ \theta_{td} \geq 0. \qquad (2)$$

It follows from Karush–Kuhn–Tucker conditions that the local maximum (Φ, Θ) of the problem (1), (2) satisfies the following system of equations with auxiliary variables interpreted as conditional probabilities $p_{tdw} = p(t \,|\, d, w)$ [33]:

$$p_{tdw} = \operatorname*{norm}_{t \in T}\bigl(\phi_{wt}\theta_{td}\bigr); \qquad (3)$$

$$n_{wt} = \sum_{d \in D} n_{dw} p_{tdw}; \quad \phi_{wt} = \operatorname*{norm}_{w \in W}\left(n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right); \quad (4)$$

$$n_{td} = \sum_{w \in d} n_{dw} p_{tdw}; \quad \theta_{td} = \operatorname*{norm}_{t \in T}\left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right); \qquad (5)$$

where the "norm" operator transforms a real vector $(x_t)_{t\in T}$ to a vector $(\tilde{x}_t)_{t\in T}$ of discrete distribution:

$$\tilde{x}_t = \operatorname*{norm}_{t\in T} x_t = \frac{\max\{x_t, 0\}}{\sum\limits_{s\in T} \max\{x_s, 0\}}.$$

The system (3)–(5) can be solved by various numerical methods. In particular, the simple-iteration method is equivalent to the EM algorithm, which is typically used in practice.

Many Bayesian topic models can be considered as special cases of ARTM with different regularizers [33, 32]. For example, PLSA [12] corresponds to the absence of regularization, $R = 0$. LDA [4] corresponds to the smoothing regularizer, which minimizes the KL-divergences $KL(\alpha\|\theta_d)$ and $KL(\beta\|\phi_t)$ for fixed distributions β, α. The choice of these distributions as uniform corresponds to the use of symmetric Dirichlet priors in Bayesian approach.

Due to the additivity ARTM can build topic models for various applications simply by choosing a suitable combination of predefined regularizers from a user extendable library. For example, a combination of five regularizers improves the interpretability of topics in [32]. Authors split the set of topics into two subsets: $T = S \sqcup B$. Domain-specific topics $t \in S$ contain terms of particular domain areas. Background topics $t \in B$ contain common words that may occur in many documents. To make topics from S sparse they maximize the KL-divergences $KL(\alpha\|\theta_d)$ and $KL(\beta\|\phi_t)$ as regularizers. To make topics from B smooth they minimize the KL-divergences $KL(\alpha\|\theta_d)$ and $KL(\beta\|\phi_t)$. Also they introduce the covariance regularizer to make all topics weakly correlated as columns of Φ matrix. The additive combination of regularizers is summarized as follows:

$$\begin{aligned} R(\Phi,\Theta) = &- \beta_0 \sum_{t\in S}\sum_{w\in W} \beta_w \ln\phi_{wt} - \alpha_0 \sum_{d\in D}\sum_{t\in S} \alpha_t \ln\theta_{td} \\ &+ \beta_1 \sum_{t\in B}\sum_{w\in W} \beta_w \ln\phi_{wt} + \alpha_1 \sum_{d\in D}\sum_{t\in B} \alpha_t \ln\theta_{td} \\ &- \gamma \sum_{t\in T}\sum_{s\in T\setminus t}\sum_{w\in W} \phi_{wt}\phi_{ws}, \end{aligned}$$

where β_0, α_0, β_1, α_1, γ are regularization coefficients. This approach has been extended in [33] by adding a regularizer that maximizes the KL-divergence $KL\big(\frac{1}{|T|}\|p(t)\big)$. This regularizer has an effect of topic selection. Starting with excessively high number of topics it eliminates insignificant, duplicated, and linearly dependent topics [34]. Compared with Hierarchical Dirichlet Process, HDP [29], topic sparsing regularizer has several advantages. It gives a more stable number of topics, it is much faster, and it can be easily combined with other topic models via additive regularization.

Another important issue for ARTM is optimization of regularization coefficients ρ_i. According to the Tikhonov's theory of ill-posed inverse problems [30], the regularization coefficients must tend to zero with the number of iteration. In this case we can achieve a stable solution. In practice, we choose the regularization path by adaptive tuning of regularization coefficients proposed in [32, 33, 34]. This is an empirical technique based on visual control of multiple intrinsic and extrinsic performance measures as functions of the number of iteration.

3. MULTIMODAL ARTM

Now assume that a document can contain not only words, but also terms of other modalities. Each modality is defined by a finite set (vocabulary) of terms W^m, $m = 1,\dots,M$. The sets W^m are disjoint.

Examples of not-word modalities are: authors, class or category labels, date-time stamps, references to/from other documents/authors, named entities, objects found in the images associated with the documents, users that read or downloaded documents, advertising banners, etc.

As in the previous section, the collection is considered to be a sample of i.i.d. triples $(w_i, d_i, t_i) \sim p(w, d, t)$ drawn from the finite probability space $W \times D \times T$, but now $W = W^1 \sqcup \dots \sqcup W^M$ is a disjoint union of the vocabularies across all modalities.

Following the idea of Correspondence LDA [3] and Dependency LDA [25] we introduce a topic model $p(w\,|\,d)$ for each modality W^m, $m = 1,\dots,M$:

$$p(w\,|\,d) = \sum_{t\in T} p(w\,|\,t)\,p(t\,|\,d) = \sum_{t\in T} \phi_{wt}\theta_{td},\ w \in W^m.$$

Stochastic matrices $\Phi^m = \big(\phi_{wt}\big)_{W^m\times T}$ of *term probabilities for the topics*, if stacked vertically, form a $W{\times}T$-matrix Φ.

To learn parameters Φ^m, Θ from the multimodal collection we maximize the log-likelihood for each m-th modality:

$$\mathscr{L}_m(\Phi^m,\Theta) = \sum_{d\in D}\sum_{w\in W^m} n_{dw} \ln p(w\,|\,d) \to \max_{\Phi^m,\Theta},$$

where n_{dw} is the number of occurrences of the term $w \in W^m$ in the document d. Note that topic distributions of documents Θ are common for all modalities.

In ARTM we add a weighted sum of regularization criteria $R(\Phi,\Theta)$ to the log-likelihood and solve a constrained multicriteria optimization problem:

$$\sum_{m=1}^{M} \tau_m \mathscr{L}_m(\Phi^m,\Theta) + R(\Phi,\Theta) \to \max_{\Phi,\Theta}; \tag{6}$$

$$\sum_{w\in W^m} \phi_{wt} = 1,\ \phi_{wt} \ge 0; \qquad \sum_{t\in T} \theta_{td} = 1,\ \theta_{td} \ge 0; \tag{7}$$

where *regularization coefficients* τ_m are used to balance the importance of different modalities. The local maximum (Φ,Θ) of the problem (6), (7) satisfies the following system of equations with auxiliary variables $p_{tdw} = p(t\,|\,d,w)$:

$$p_{tdw} = \operatorname*{norm}_{t\in T}\big(\phi_{wt}\theta_{td}\big); \tag{8}$$

$$n_{wt} = \sum_{d\in D} \tau_{m(w)} n_{dw} p_{tdw};$$

$$n_{td} = \sum_{w\in d} \tau_{m(w)} n_{dw} p_{tdw};$$

$$\phi_{wt} = \operatorname*{norm}_{w\in W^m}\left(n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right); \tag{9}$$

$$\theta_{td} = \operatorname*{norm}_{t\in T}\left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right); \tag{10}$$

where $m(w)$ is the modality of the term w, $w \in W^{m(w)}$.

The system of equations (8)–(10) follows from Karush–Kuhn–Tucker conditions (see Appendix A for the proof). For single modality ($M = 1$) it gives the regularized EM algorithm described in the previous section.

Many previous topic models for labeled documents can be considered as specials cases of multimodal ARTM. Most of them are based on LDA model and use Dirichlet priors, which correspond to smoothing regularization. From ARTM perspective, there is little reason to always use only the smoothing regularizer.

Following topic models exactly correspond to the multimodal ARTM, up to the modality sense. A topic model of document content and hypertext connectivity [6] has the modality of documents to which a given document has a hyperlink. The Conditionally Independent LDA, CI-LDA [21] has the modality of named entities mentioned in a given document. The Tag-LDA [27] has the modality of tags as special kind of words. The LDA-JS and LDA-post [9] has the modality of publications cited in a given document; an additional regularizer takes into account that cited documents are likely to share similar topics. Both models are designed to estimate the strength of influence of cited publications. The Dependency LDA [25] has the modality of document categories or class labels. The MultiLingual LDA, ML-LDA [22] and the PolyLingual Topic Model, PLTM [19] have L modalities for L different languages; parallel documents always share one identical topic distribution. The BiLingual LDA, BiLDA [8] is also a multilanguage topic model, but the number of modalities is restricted by two.

4. ONLINE PARALLEL EM-ALGORITHM

Like Online LDA [11] and Online PLSA [1] we split the collection D into batches D_b, $b = 1, \dots, B$, and organize EM iterations so that each document vector θ_d is iterated until convergence at a constant matrix Φ, see Algorithm 1 and 2. Matrix Φ is updated rarely, after all documents from the batch are processed. For a large collection matrix Φ often stabilizes after small initial part of the collection. Therefore a single pass through the collection might be sufficient to learn a topic model. The second pass may be needed for the initial part of the collection.

The online reorganization of the EM iterations is not necessarily associated with Bayesian inference used in [11]. Different topic models, from PLSA to multimodal and regularized models, can be learned by the above online EM algorithm.

Algorithm 1 does not specify how often to synchronize Φ matrix at steps 5–8. It can be done after every batch or less frequently (for instance if $\frac{\partial R}{\partial \phi_{wt}}$ takes long time to evaluate). This flexibility is especially important for concurrent implementation of the algorithm, where multiple batches are processed in parallel. In this case synchronization can be triggered when a fixed number of documents had been processed since the last synchronization.

Each D_b batch is stored on disk in a separate file, and only a limited number of batches is loaded into the main memory at any given time. The entire Θ matrix is also never stored in the memory. As a result, the memory usage stays constant regardless of the size of the collection.

Algorithm 1: Online EM-algorithm for multimodal ARTM

```
Input: collection $D_b$, discounting factor $\rho \in (0, 1]$;
Output: matrix $\Phi$;
1 initialize $\phi_{wt}$ for all $w \in W$ and $t \in T$;
2 $n_{wt} := 0$, $\tilde{n}_{wt} := 0$ for all $w \in W$ and $t \in T$;
3 for all batches $D_b$, $b = 1, \dots, B$
4     $(\tilde{n}_{wt}) := (\tilde{n}_{wt}) + \mathsf{ProcessBatch}(D_b, \Phi)$;
5     if (synchronize) then
6         $n_{wt} := \rho n_{wt} + \tilde{n}_{dw}$ for all $w \in W$ and $t \in T$;
7         $\phi_{wt} := \mathop{\mathrm{norm}}_{w \in W^m} \bigl(n_{wt} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}}\bigr)$ for all $w \in W^m$,
          $m = 1, \dots, M$ and $t \in T$;
8         $\tilde{n}_{wt} := 0$ for all $w \in W$ and $t \in T$;
```

Algorithm 2: $\mathsf{ProcessBatch}(D_b, \Phi)$

```
Input: batch $D_b$, matrix $\Phi = (\phi_{wt})$;
Output: matrix $(\tilde{n}_{wt})$;
1 $\tilde{n}_{wt} := 0$ for all $w \in W$ and $t \in T$;
2 for all $d \in D_b$
3     initialize $\theta_{td} := \frac{1}{|T|}$ for all $t \in T$;
4     repeat
5         $p_{tdw} := \mathop{\mathrm{norm}}_{t \in T} \bigl(\phi_{wt}\theta_{td}\bigr)$ for all $w \in d$ and $t \in T$;
6         $n_{td} := \sum_{w \in d} \tau_{m(w)} n_{dw} p_{tdw}$ for all $t \in T$;
7         $\theta_{td} := \mathop{\mathrm{norm}}_{t \in T} \bigl(n_{td} + \theta_{td} \frac{\partial R}{\partial \theta_{td}}\bigr)$ for all $t \in T$;
8     until $\theta_d$ converges;
9     $\tilde{n}_{wt} := \tilde{n}_{wt} + \tau_{m(w)} n_{dw} p_{tdw}$ for all $w \in d$ and $t \in T$;
```

To split collection into batches and process them concurrently is a common approach, introduced in AD-LDA algorithm [20], and then further developed in PLDA [36] and PLDA+ [17] algorithms. These algorithms require all concurrent workers to become idle before an update of the Φ matrix. Such synchronization step adds a large overhead in the online algorithm where Φ matrix is updated multiple times on each iteration. An alternative architecture without the synchronization step is described in [28], however it mostly targets a distributed cluster environment. In our work we develop an efficient single-node architecture where all workers benefit from the shared memory space.

To run multiple ProcessBatch in parallel the inputs and outputs of this routine are stored in two separate in-memory queues, locked for push and pop operations with spin locks. This approach does not add any noticeable synchronization overhead because both queues only store smart pointers to the actual data objects, so push and pop operations does not involve copying or relocating big objects in the memory.

Smart pointers are also essential for lifecycle of the Φ matrix. This matrix is *read* by all processors threads, and can be *written* at any time by the merger thread. To update Φ without pausing all processor threads we keep two copies — an *active* Φ and a *background* Φ matrices. The active matrix is read-only, and is used by the processor threads. The background matrix is being built in a background by the merger thread at steps 6 and 7 of Algorithm 1, and once it is ready merger thread marks it as active. Before pro-

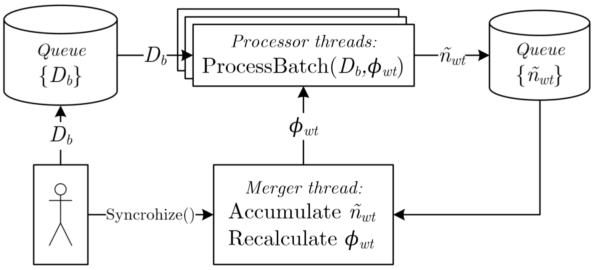

Figure 1: Diagram of parallelization components

cessing a new batch the processor thread gets the current active matrix from the merger thread. This object is passed via shared smart pointer to ensure that processor thread can keep ownership of its Φ matrix until the batch is fully processed. As a result, all processor threads keep running concurrently with the update of Φ matrix.

All processor threads share the same Φ matrix, which means that memory usage stays at constant level regardless of how many cores are used for computation. Using memory for two copies of the Φ matrix in our opinion gives a reasonable usage balance between memory and CPU resources. An alternative solution with only one Φ matrix is also possible, but it would require a heavy usage of atomic CPU instructions. Such operations are very efficient, but still come at a considerable synchronization cost,[1] and using them for all reads and writes of the Φ matrix would cause a significant performance degradation for merger and processor threads. Besides, an arbitrary overlap between reads and writes of the Φ matrix eliminates any possibility of producing a deterministic result. The design with two copies of the Φ matrix gives much more control over this and in certain cases allows the algorithm to behave in a fully deterministic way.

The design with two Φ matrices only supports a single merger thread, and we believe it should handle all $\tilde{n}_{wt}$ updates coming from many threads. This is a reasonable assumption because merging at step 6 takes only about $O(|W| \cdot |T|)$ operations to execute, while ProcessBatch takes $O(n|T|I)$ operations, where n is the number of non-zero entries in the batch, I is the average number of inner iterations in ProcessBatch routine. The ratio $n/|W|$ is typically from 100 to 1000 (based on datasets in UCI Bag-Of-Words repository), and I is $10 \ldots 20$, so the ratio safely exceeds the expected number of cores (up to 32 physical CPU cores in modern workstations, and even 60 cores of the Intel Xeon Phi co-processors).

We use dense single-precision matrices to represent Φ and Θ. Together with the Φ matrix we store a global dictionary of all terms $w \in W$. This dictionary is implemented as std::unordered_map that maps a string representation of $w \in W$ into its integer index in the Φ matrix. This dictionary can be extended automatically as more and more batches came through the system. To achieve this each batch D_b contains a local dictionary W_b, listing all terms that occur in the batch. The n_{dw} elements of the batch are stored as a sparse CSR matrix (Compressed Sparse Raw format), where each row correspond to a document $d \in D_b$, and terms w run over a local batch dictionary W_b.

For performance reasons Φ matrix is stored in column-major order, and Θ in row-major order. This layout ensures that $\sum_t \phi_{wt}\theta_{td}$ sum runs on contiguous memory blocks. In both matrices all values smaller than 10^{-16} are always replaced with zero to avoid performance issues with denormalized numbers.[2]

The parallel online EM-algorithm for multimodal ARTM is implemented in BigARTM open source project available from `http://bigartm.org` under the New BSD License. The core of the library is written in C++ and is exposed via two equally rich APIs for C++ and Python. The library is cross-platform and can be built for Linux, Windows and OS X in both 32 and 64 bit configuration.

5. EXPERIMENTS

Runtime performance. In first experiment we evaluate the runtime performance and intrinsic quality of BigARTM against two popular software packages — Gensim [24] and Vowpal Wabbit.[3]

All three libraries (VW.LDA, Gensim and BigARTM) work out-of-core, e. g. they are designed to process data that is too large to fit into a computer's main memory at one time. This allowed us to benchmark on a fairly large collection — 3.7 million articles from the English Wikipedia.[4] The conversion to bag-of-words was done with gensim.make_wikicorpus script,[5] which excludes all non-article pages (such as category, file, template, user pages, etc), and also pages that contain less than 50 words. The dictionary is formed by all words that occur in at least 20 documents, but no more than in 10% documents in the collection. The resulting dictionary was caped at $|W| = 100\,000$ most frequent words. Perplexity is used as an intrinsic quality measure:

$$\mathscr{P}(D,p) = \exp\biggl(-\frac{1}{n}\sum_{d\in D}\sum_{w\in d} n_{dw} \ln p(w \mid d)\biggr). \qquad (11)$$

Vowpal Wabbit (VW) is a library of online algorithms that cover a wide range of machine learning problems. For topic modeling VW has the VW.LDA algorithm, based on the Online Variational Bayes LDA [11]. VW.LDA is neither multi-core nor distributed, but an effective single-threaded implementation in C++ made it one of the fastest tools for topic modeling.

Gensim library specifically targets the area of topic modeling and matrix factorization. It has two LDA implementations — LdaModel and LdaMulticore, both based on the same algorithm as VW.LDA (Online Variational Bayes LDA [11]). Gensim is entirely written in Python. Its high

[1] `http://stackoverflow.com/questions/2538070/atomic-operation-cost`

[2] `http://en.wikipedia.org/wiki/Denormal_number#Performance_issues`

[3] `https://github.com/JohnLangford/vowpal_wabbit/`

[4] `http://dumps.wikimedia.org/enwiki/20141208/`

[5] `https://github.com/piskvorky/gensim/tree/develop/gensim/scripts/`

Table 1: The comparison of BigARTM with VW.LDA and Gensim; *train* is the time for model training, *inference* is the time for calculation of θ_d of $100\,000$ held-out documents, *perplexity* is calculated according to (11) on held-out documents.

library	procs	train	inference	perplexity
BigARTM	1	35 min	72 sec	4000
LdaModel	1	369 min	395 sec	4161
VW.LDA	1	73 min	120 sec	4108
BigARTM	4	9 min	20 sec	4061
LdaMulticore	4	60 min	222 sec	4111
BigARTM	8	4.5 min	14 sec	4304
LdaMulticore	8	57 min	224 sec	4455

performance is achieved through the usage of NumPy library, built over low-level BLAS libraries (such as Intel MKL, ATLAS, or OpenBLAS). In LdaModel all batches are processed sequentially, and the concurrency happens entirely within NumPy. In LdaMulticore the workflow is similar to BigARTM — several batches are processed concurrently, and there is a single aggregation thread that asynchronously merges the results.

Table 1 compares the performance of VW.LDA, Gensim LdaModel and LdaMulticore (0.10.3 under Python 2.7), and BigARTM for Amazon AWS c3.8xlarge with 32 virtual cores over Intel-based CPU with 16 physical cores with hyper-threading. Each run performs one pass over the Wikipedia corpus and produces a model with $|T| = 100$ topics. The collection was split into batches with 10K documents each (`chunksize` in Gensim, `minibatch` in VW.LDA). The update rule in online algorithm used the discounting factor $\rho = (b + \tau_0)^{-0.5}$, where b is the number of batches processed so far, and τ_0 is a constant offset parameter introduced in [11], in our experiment $\tau_0 = 64$. Updates were performed after each batch in non-parallel runs, and after P batches when running in P threads. To make fair comparison we used only smoothing regularization in BigARTM, which is equivalent to the LDA model. LDA priors were fixed as $\alpha = 0.1$, $\beta = 0.1$ for all models.

Combination of regularizers. BigARTM has a built-in library of regularizers, which can be used in any combination. In the following experiment we combine regularizers described in section 2: sparsing of ϕ_t, sparsing of θ_d, and pairwise decorrelation of ϕ_t distributions. This combination improves several quality measures without significant loss of perplexity for the offline implementation of ARTM [33]. The goal of our experiment is to show that this remains true for the online implementation in BigARTM. We use the following built-in performance measures: the hold-out perplexity, the sparsity of Φ and Θ matrices, and the characteristics of topic lexical kernels (size, purity, and contrast) averaged across all topics. Table 2 compares the results of additive combination of regularizers (ARTM) and the usual LDA model. Figure 2 presents performance measures as functions of the number of processed documents. The first chart shows perplexity and sparsity of Φ, Θ matrices, and the second chart shows average lexical kernel measures.

Table 2: Comparison of LDA and BigARTM models: $\mathcal{P}_{10k}$, $\mathcal{P}_{100k}$ — hold-out perplexity on 10K and 100K documents sets, $\mathcal{S}_\Phi$, $\mathcal{S}_\Theta$ — sparsity of Φ and Θ matrices (in %), $\mathcal{K}_s$, $\mathcal{K}_p$, $\mathcal{K}_c$ — average topic kernel size, purity and contrast respectively.

Model	$\mathcal{P}_{10k}$	$\mathcal{P}_{100k}$	$\mathcal{S}_\Phi$	$\mathcal{S}_\Theta$	$\mathcal{K}_s$	$\mathcal{K}_p$	$\mathcal{K}_c$
LDA	3436	3801	0.0	0.0	873	0.533	0.507
ARTM	3577	3947	96.3	80.9	1079	0.785	0.731

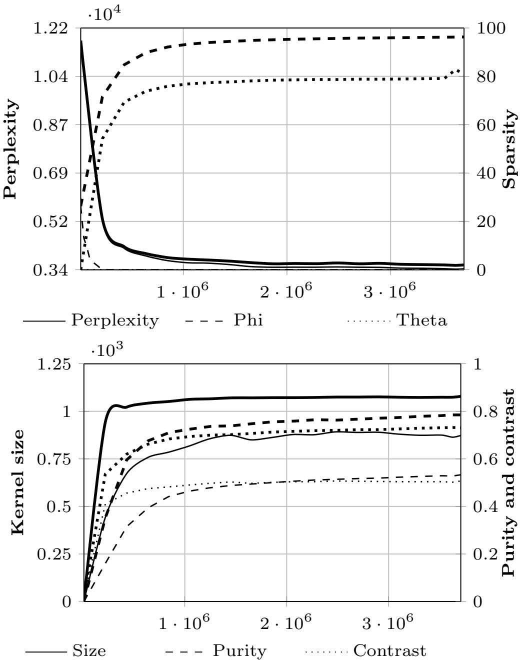

Figure 2: Comparison of LDA (thin) and ARTM (bold) models. The number of processed documents is shown along the X axis.

Text classification. Support vector machine (SVM) based on token frequencies is known to be one of the best methods for text classification. However, according to [25] topic models demonstrate even better quality in case of unbalanced interdependent and intersecting classes. Our experiment aims at showing that multimodal regularized topic models in BigARTM are at least no worse than Dependency LDA from [25]. Dependency LDA is in fact a multimodal topic model with two modalities: words and class labels.

The EUR-lex collection contains about 20K documents split into train and test sets to provide the reproducibility of the results [25]. The original size of the dictionary is over 190K tokens. Preprocessing from [25] removes all tokens encountered less than 20 times, and reduces the dictionary to about 20K tokens. Class labels, encountered only once, are also removed to result in about 3250 classes. Each document might belong to several classes.

For both Dependency LDA and ARTM the label regularization [25] was used. The quality measures in our experiment are as follows: AUC$_{PR}$ — the area under the precision-recall curve; AUC — the area under ROC-curve; OneErr — the

Table 3: Multimodal ARTM, Dependency LDA and SVM classification models. The best results are in bold. T_{opt} is the optimal number of topics.

	T_{opt}	AUC$_{PR}\uparrow$	AUC$\uparrow$	OneErr$\downarrow$	IsErr$\downarrow$
ARTM	15 000	**0.529**	0.980	**27.1**	**94.2**
DLDA	200	0.492	**0.982**	32.0	97.2
SVM	–	0.435	0.975	31.6	98.1

ratio of documents with the most probable label not from the correct set; IsErr — the ratio of documents with not ideal classification.

The results are provided in Table 3. ARTM performs better than both Dependency LDA and SVM by the three measures out of four. It is interesting to note that while the number of topics increases up to 15 000, ARTM provides better classification quality, while the optimal number of topics for Dependency LDA is 200.

Cross-language search. The following experiment shows that multimodal topic model may be used as multilingual one, with languages of parallel texts treated as modalities. The experiment was held on the EuroParl collection [15] of European Parliament Proceedings. Proceedings in English and Spanish were chosen, as these languages are often used for multilingual topic model comparison. As in [19, 23, 18], a single document is a speech of one speaker at one session.

We measure the quality of cross-language search by precision, i. e. the fraction of query documents q, for which their translations d are the closest by Hellinger distance:

$$H^2(d,q) = \frac{1}{2}\sum_{t\in T}\bigl(\sqrt{p(t\,|\,d)} - \sqrt{p(t\,|\,q)}\bigr)^2.$$

Training set includes proceedings from 1996 to 1999, and from 2001 to 2002, test set includes proceedings of 2000 and the first 9 months of 2003. The same partitioning is used in [23] and [18]. Moreover, as in [19, 18], the test comprised documents of the length more or equal than 100 words. The total number of documents is 67379 in the training set, and 16068 in the test set. Built-in capability of BigARTM to filter the dictionary was used: all rare words, that appear in less than 20 documents, and stop-words, that appear in more than 50% of documents, were discarded. Table 4 shows the comparison of models from [19, 23, 18] and our ARTM. For the first two models, the authors provide search precision for 50 topics only. ARTM performs slightly worse than JPLSA, but we note, that one iteration of BigARTM takes 30 seconds for 50 topics and 40 seconds for 100 topics, while one iteration of JPLSA takes 31 minutes. ARTM performs better if compared with models from [18].

Table 4: Cross-language search precision for different models. The best value in each column is bolded.

	Number of topics T			
Model	50	100	200	500
PLTM [19]	0.812	–	–	–
JPLSA [23]	**0.989**	–	–	–
PLTM-He [18]	0.943	0.985	0.994	0.993
PLTM-He kd-trees [18]	0.949	0.989	0.995	0.996
BigARTM	0.972	**0.990**	**0.996**	**0.997**

Table 5: The quality of recommendations for baseline matrix factorization model, unimodal model with only modality of user likes, and two multimodal models incorporating words and user-specified data (tags and categories).

Model	Recall@5	Recall@10	Recall@20
baseline [13]	0.591	0.652	0.678
likes	0.62	0.59	0.65
likes + words	0.79	0.64	0.68
all modalities	**0.80**	**0.71**	**0.69**
no regularization	0.79	0.71	0.68

Recommending articles of collective blog. Here we describe how multimodal topic modeling can be used for recommending articles in a collective blog. Collective blog is an on-line platform where users can publish articles and respond to the articles of other authors. To make recommendations we add user's positive feedback to the article as a modality. For the experiment we used dataset of about 130K articles with user feedback from `http://habrahabr.ru` — the most popular IT-oriented social blogging platform in Russia. The articles from our dataset have five modalities: words from text, users who liked articles, authors, tags and categories (hubs) specified by users.

To construct list of recommended articles to the user u we estimate his topic distribution $p(t\,|\,u)$ and rank documents according to $p(d\,|\,u)$. To assess the quality of recommendations we split the set of user–article interactions (likes) on two disjoint subsets in proportion $1:1$, the former subset is used for estimating user topics and the latter subset contains hold-out preferences used to compute Recall@k metric (the proportion of liked articles among top k recommendations). As a baseline recommendation model we used weighted regularized matrix factorization [13] based on user likes. This approach is commonly used in recommender systems.

Table 5 presents the results of a comparison of three models. Performance of the topic models is comparable or better than baseline. Additional modalities improves recommendation ranking significantly. The combination of all modalities with regularizers of sparsity and decorrelation does not degrade the quality of recommendation but provides much more sparse and interpretable model. It is well known that factors of Weighted Matrix Factorization are dense and their components do not correspond to human-sensible topics. By using regularizers we could make interpretability of factors even better. The interpretability of the user profile $p(t\,|\,u)$ enables new ways of using recommendation model.

6. CONCLUSIONS

Additive Regularization of Topic Models (ARTM) is a powerful non-Bayesian framework, which facilitates the development, inference, combining, and understanding of topic models. Combining multiple modalities with multiple regularization criteria covers dozens of models previously studied only in the Bayesian settings. The inference is reduced to the differentiation of each regularizer with respect to the model parameters and then applying of the ready-to-use formula of the regularized M-step.

BigARTM is an open source project for parallel online multimodal regularized topic modeling of large text collections. Its implementation is much faster than existing popular topic modeling tools. BigARTM provides a high flexibility for various applications due to multimodality and additive combinations of regularizers.

BigARTM architecture has a rich potential. Current components can be reused in a distributed solution that runs on cluster. Further improvement of single-node can be achieved by offloading batch processing into GPU.

Acknowledgements.

The work was supported by the Russian Foundation for Basic Research (14-07-00847, 14-07-00908, 14-07-31176) and by Skolkovo Institute of Science and Technology (081-R).

7. REFERENCES

[1] N. Bassiou and C. Kotropoulos. Online PLSA: Batch updating techniques including out-of-vocabulary words. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(11):1953–1966, Nov 2014.

[2] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[3] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 127–134, New York, NY, USA, 2003.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] J.-T. Chien and Y.-L. Chang. Bayesian sparse topic model. *Journal of Signal Processessing Systems*, 74:375–389, 2013.

[6] D. A. Cohn and T. Hofmann. The missing link — a probabilistic model of document content and hypertext connectivity. In *NIPS*, pages 430–436, 2000.

[7] A. Daud, J. Li, L. Zhou, and F. Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of Computer Science in China*, 4(2):280–301, 2010.

[8] W. De Smet and M.-F. Moens. Cross-language linking of news stories on the web using interlingual topic modelling. In *Proceedings of the 2Nd ACM Workshop on Social Web Search and Mining*, SWSM '09, pages 57–64, New York, NY, USA, 2009.

[9] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 233–240, New York, NY, USA, 2007.

[10] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *ICML'11*, pages 1041–1048, 2011.

[11] M. D. Hoffman, D. M. Blei, and F. R. Bach. Online learning for latent dirichlet allocation. In *NIPS*, pages 856–864. Curran Associates, Inc., 2010.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[13] Y. Hu, Y. Koren and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *IEEE ICDM'08*. 2008.

[14] O. Khalifa, D. Corne, M. Chantler, and F. Halley. Multi-objective topic modelling. In *7th International Conference Evolutionary Multi-Criterion Optimization (EMO 2013)*, pages 51–65. Springer LNCS, 2013.

[15] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79-86, Phuket, Thailand, 2005.

[16] M. O. Larsson and J. Ugander. A concave regularization technique for sparse mixture models. In *Advances in Neural Information Processing Systems 24*, pages 1890–1898, 2011.

[17] Z. Liu, Y. Zhang, E. Y. Chang, and M. Sun. PLDA+: parallel latent Dirichlet allocation with data placement and pipeline processing. *ACM Trans. Intell. Syst. Technol.*, 2(3):26:1–26:18, May 2011.

[18] D. Mimno, M. Hoffman, and D. Blei. Sparse stochastic inference for latent Dirichlet allocation. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1599–1606, 2012.

[19] D. Mimno, H. M. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, EMNLP '09, pages 880–889, 2009.

[20] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 10:1801–1828, Dec. 2009.

[21] D. Newman, C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 680–686, New York, NY, USA, 2006.

[22] X. Ni, J.-T. Sun, J. Hu, and Z. Chen. Mining multilingual topics from wikipedia. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1155–1156, 2009.

[23] J. C. Platt, K. Toutanova, W.-T. Yih. Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 251–261, Stroudsburg, PA, USA, 2010.

[24] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010.

[25] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2), pages 157–208, 2012.

[26] M. Shashanka, B. Raj, and P. Smaragdis. Sparse

overcomplete latent variable decomposition of counts data. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems, NIPS-2007*, pages 1313–1320. MIT Press, Cambridge, MA, 2008.

[27] X. Si and M. Sun. Tag-lda for scalable real-time tag recommendation. *Journal of Information & Computational Science*, 6:23–31, 2009.

[28] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Proc. VLDB Endow.*, 3(1-2):703–710, Sept. 2010.

[29] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[30] A. N. Tikhonov, V. Y. Arsenin. Solution of ill-posed problems. W. H. Winston, Washington, DC. 1977.

[31] K. V. Vorontsov. Additive regularization for topic models of text collections. *Doklady Mathematics*, 89(3):301–304, 2014.

[32] K. V. Vorontsov and A. A. Potapenko. Additive regularization of topic models. *Machine Learning, Special Issue on Data Analysis and Intelligent Optimization*, 2014.

[33] K. V. Vorontsov and A. A. Potapenko. Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization. In *AIST'2014, Analysis of Images, Social networks and Texts*, volume 436, pages 29–46. Springer International Publishing Switzerland, Communications in Computer and Information Science (CCIS), 2014.

[34] K. V. Vorontsov, A. A. Potapenko, and A. V. Plavin. Additive Regularization of Topic Models for Topic Selection and Sparse Factorization. In *3rd Int'l Symposium On Learning And Data Sciences (SLDS 2015)*, Royal Holloway, University of London, UK. Springer, LNAI 9047, pages 193–202, 2015.

[35] C. Wang and D. M. Blei. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In *NIPS*, pages 1982–1989. Curran Associates, Inc., 2009.

[36] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang. PLDA: Parallel latent Dirichlet allocation for large-scale applications. In *Proceedings of the 5th International Conference on Algorithmic Aspects in Information and Management*, pages 301–314, 2009.

Appendix

Consider the system of equations (8)–(10).

Topic t is called *regular* for modality m if $n_{wt}+\phi_{wt}\frac{\partial R}{\partial \phi_{wt}} > 0$ for at least one term $w \in W^m$. If the reverse inequality holds for all $w \in W^m$ then topic t is called *irregular*; in this case the t-th vector-column in matrix Φ^m equals zero and can not represent a discrete distribution. This means that topic t for the modality m must be excluded from the model. This mechanism can be used to eliminate irrelevant topics and determine the number of topics.

Document d is called *regular* if $n_{td}+\theta_{td}\frac{\partial R}{\partial \theta_{td}} > 0$ for at least one topic $t \in T$. If the reverse inequality holds for all $t \in T$ then document d is called *irregular*; in this case the d-th vector-column in matrix Θ equals zero and can not represent a discrete distribution. This means that document d must be excluded from the model. For example, a document may be too short or irrelevant to the given collection.

THEOREM 1. *If the function $R(\Phi,\Theta)$ is continuously differentiable and (Φ,Θ) is the local maximum of the problem* (6), (7) *then for any regular topic t and any regular document d the system of equations* (8)–(10) *holds.*

PROOF. For the local minimum Φ^m,Θ of the problem (6), (7) the Karush–Kuhn–Tucker (KKT) conditions can be written as follows:

$$\sum_d n_{dw}\frac{\theta_{td}}{p(w\,|\,d)} + \frac{\partial R}{\partial \phi_{wt}} = \lambda_t - \lambda_{wt};$$

$$\lambda_{wt} \ge 0; \quad \lambda_{wt}\phi_{wt} = 0;$$

$$\sum_m \tau_m \sum_{w\in W^m} n_{dw}\frac{\phi_{wt}}{p(w\,|\,d)} + \frac{\partial R}{\partial \theta_{td}} = \mu_d - \mu_{td};$$

$$\mu_{td} \ge 0; \quad \mu_{td}\theta_{td} = 0;$$

where λ_t, λ_{wt}, μ_d, μ_{td} are KKT multipliers for normalization and nonnegativity constraints.

Let us multiply both sides of the first equation by ϕ_{wt}, both sides of the second equation by θ_{td}, and reveal the auxiliary variable p_{tdw} from (8) in the left-hand side of both equations. Then we sum the right-hand side of the first equation over d, the right-hand side of the second equation over t:

$$\phi_{wt}\lambda_t = \sum_d n_{dw}\frac{\phi_{wt}\theta_{td}}{p(w\,|\,d)} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}} = n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}};$$

$$\theta_{td}\mu_d = \sum_m \tau_m \sum_{w\in W^m} n_{dw}\frac{\phi_{wt}\theta_{td}}{p(w\,|\,d)} + \theta_{td}\frac{\partial R}{\partial \theta_{td}} = n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}.$$

An assumption that $\lambda_t \le 0$ contradicts the regularity condition for the (t,m) pair. Then $\lambda_t > 0$. Either $\phi_{wt} = 0$ or both sides of the first equation are positive. Combining these two cases in one formula, we write:

$$\phi_{wt}\lambda_t = \max\left\{n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}, 0\right\}. \tag{12}$$

Analogously, an assumption that $\mu_d \le 0$ contradicts the regularity condition for the document d. Then $\mu_d > 0$. Either $\theta_{td} = 0$ or both sides of the second equation are positive, consequently,

$$\theta_{td}\mu_d = \max\left\{n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}, 0\right\}. \tag{13}$$

Let us sum both sides of the first equation over all $w \in W^m$, then both sides of the second equation over all $t \in T$:

$$\lambda_t = \sum_{w\in W^m}\max\left\{n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}, 0\right\}; \tag{14}$$

$$\mu_d = \sum_{t\in T}\max\left\{n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}, 0\right\}. \tag{15}$$

Finally, we obtain (9) and (10) by expressing ϕ_{wt} from (12) and (14), then by expressing θ_{td} from (13) and (15). □